J Real-Time Image Proc
DOI 10.1007/s11554-014-0406-1

ORIGINAL RESEARCH PAPER

Adaptive pattern recognition in real-time video-based soccer analysis

Marc Schlipsing · Jan Salmen · Marc Tschentscher · Christian Igel

Received: 16 October 2013 / Accepted: 4 February 2014

Abstract Computer-aided sports analysis is demanded by coaches and the media. Image processing and machine learning techniques that allow for "live" recognition and tracking of players exist. But these methods are far from collecting and analyzing event data fully autonomously. To generate accurate results, human interaction is required at different stages including system setup, calibration, supervision of classifier training, and resolution of tracking conflicts. Furthermore, the real-time constraints are challenging: in contrast to other object recognition and tracking applications, we cannot treat data collection, annotation, and learning as an offline task. A semi-automatic labeling of training data and robust learning given few examples from unbalanced classes are required. We present a real-time system acquiring and analyzing video sequences from soccer matches. It estimates each player's position throughout the whole match in real-time. Performance measures derived from these raw data allow for an objective evaluation of physical and tactical profiles of teams and individuals. The need for precise object recognition, the restricted working environment, and the technical limitations of a mobile setup are taken into account. Our contribution is twofold: (1) the deliberate use of machine learning and pattern recognition techniques allows us to achieve high classification accuracy in varying environments. We systematically evaluate combinations of image features and learning machines in the given online scenario. Switching between classifiers depending on the amount of training data and available training time improves robustness and efficiency. (2) A proper human–machine interface decreases the number of required operators who are incorporated into the system's learning process. Their main task reduces to the identification of players in uncertain situations. Our experiments showed high performance in the classification task achieving an average error rate of 3 % on three real-world datasets. The system was proved to collect accurate tracking statistics throughout different soccer matches in real-time by incorporating two human operators only. We finally show how the resulting data can be used instantly for consumer applications and discuss further development in the context of behavior analysis.

Keywords Sports analysis · Supervised learning · Motion analysis · Human–machine interfaces

1 Introduction

Computer vision and image analysis are becoming more and more important in sports analytics, the science of analyzing and modeling processes underlying sporting events. Sports with high media coverage create a demand for systematic review and objective evaluation of the performance of individual athletes as well as of teams. Across almost all sports, management and coaches make use of statistics and categorized video material to support their strategies.

Electronic supplementary material The online version of this article (doi:10.1007/s11554-014-0406-1) contains supplementary material, which is available to authorized users.

M. Schlipsing (✉) · J. Salmen · M. Tschentscher
Institut für Neuroinformatik, Ruhr-Universität Bochum,
44780 Bochum, Germany
e-mail: marc.schlipsing@ini.rub.de

C. Igel
Department of Computer Science, University of Copenhagen,
2100 Copenhagen, Denmark
e-mail: igel@diku.dk

Published online: 25 February 2014

We consider a framework for real-time analysis of soccer matches [30]. It consists of two high-definition cameras, one desktop PC, and two laptops. Our system collects positional data for each player during the whole match. These data can be accessed for various purposes such as processing for television broadcasting, mobile applications, and professional analysis. In particular, processed tracking data provide important insights for physical and tactical performance evaluation by coaches [3] as depicted in Sect. 5.5. Moreover, in contrast to commercial systems, the targeted use case is extended to tracking at any soccer field, be it training pitch, small stadiums or away matches with a system operated by two briefly trained laymen. Thus, the development of the presented framework was driven by the following design goals:

- *Mobility* One is able to quickly set up and calibrate the system at any location, be it stadium, training site, or indoor court.
- *Low cost* The hardware requirements are small, because only off-the-shelf hardware is used.
- *High degree of automation* The recognition system can be set up and run by only two human operators.
- *Accuracy* State-of-the-art pattern recognition techniques ensure accurate detection and classification performance.

In the following, we present our video-based sports analysis system in detail. We put an emphasis on the “online” training task that has to be solved for a *live* application of such a tracking system. This includes the efficient combination of unsupervised and supervised multi-category classifications and the involved human–machine interaction (HMI).

Our approach takes into account the requirements for robust object recognition and tracking, the constraint operator working environment, and the technical limitations of a mobile setup. This requires new techniques for efficient data annotation and iterative classifier training for the given scenario.

In our sample application, the classification task reduces to distinguishing different team clothing. There are five main categories: outfield players and goalkeepers of both teams and the referees. Being embedded in a real-time process, the classification module is subject to constraints regarding the choice of image features and computational complexity of the classifier. We present a comparative study that justifies our design choices for the classification module. We employed combinations of color histograms from three color spaces as a robust representation of non-rigid objects and compare their performance with principal component analysis (PCA) feature extraction and spatiograms. Moreover, different types of classifiers, namely a nearest neighbor approach, linear discriminant analysis, and two multi-class extensions of support vector machines, were evaluated.

The following sections present related work and give an overview of our recognition system. Section 4 points out the real-time constraints and their implications for feature/classifier choice, describes the proposed procedure, and states our empirical results. Section 5 discusses the HMI approach followed by a brief review of its evaluation. We finish with an overall conclusion and an outlook towards future research directions.

2 Related work

Video analysis of sports based on television broadcasts has been done to categorize the material with respect to the type of sport, the camera view [37], and interesting events like scores or offside [1, 8]. Nevertheless, due to the limited field coverage, TV material is not suitable for robustly tracking all actors involved in the game.

Approaches to player tracking based on task-specific camera setups (mainly in the context of soccer) are reviewed by Xinguo Farin [35] and D’Orazio and Leo 8]. For alternative systems based on multiple cameras distributed in the stadium we refer to Poppe et al. [24], Ben Shitrit et al. [4] and Ren et al. [27]. In the case of various camera positions within the stadium, different lighting conditions have to be considered, e.g., by a cumulative brightness transfer function [25]. Other notable publications relevant in the context of our study focus on detection and tracking using color and depth information [29], unsupervised feature extraction [23] and address the tracking task with graph representations [10]. The importance of analyzing different color spaces for the image segmentation in soccer analysis is pointed out by Xu et al. [36] and Vandenbroucke et al. [33], who introduced an adapted hybrid color space. As a state-of-the-art baseline, we considered *spatiograms*, an extension of histograms, proposed in the context of region-based object tracking [5].

None of the aforementioned approaches is able to identify players in person. They only recognize team membership. It is noteworthy that skilled humans are able to identify players in the videos, incorporating different hints like players’ physique, skin and hair color, course of motions, and position relative to the rest of the team.

First commercial systems for the analysis of soccer videos have reached the market, for instance *Tracab*,[1] *AmiscoPro*,[2] and *Vis.Track*.[3] They either use up to 16 mobile cameras and a stereo vision approach for tracking

[1] http://www.tracab.com.

[2] http://www.sport-universal.com.

[3] http://www.bundesliga-datenbank.de.

or require several permanently installed cameras. Statistics are either captured live—with the help of up to eight human operators—or offline after 48 h. None of the mentioned systems is able to operate fully autonomously.

For the given live scenario, where the visual appearance of all actors is *not* known prior to the match, no satisfying solution has been proposed yet. The standard setup for classification modules presented in the mentioned literature is an offline-learning procedure. Commercial systems counteract this issue by massive human effort, e.g., manually selecting representative colors during warm-up to initialize segmentation and classification modules. Such approaches are neither efficient nor robust.

In their review, D'Orazio and Leo conclude that "*a great deal of work should be directed towards the enhancement of automatic analysis to reduce manual intervention and improve their performance*" [8]. This study points out the machine learning part to achieve high classification accuracy in varying environments while requiring only little human intervention. Still, human interaction is required for

- system setup and calibration,
- supervision of machine learning algorithms,
- identification of individual players, and
- resolution of multi-object tracking conflicts in crowded environments.

While there are numerous studies focusing on the improvement of computer vision techniques involved in the recognition process, there is little work done in the field of efficient human–machine interaction. The goal of that interaction is not limited to complementing weaker system parts by human operators, but to incorporate the operators into the process of machine learning for ensuring an accurate and robust performance.

3 Video processing overview

The recognition system operates in real-time, allows to analyze full field views, and relies on portable, affordable hardware. Using two static high-resolution cameras we produce a panoramic image capturing the whole field. Based on this video stream we generate two cues for object segmentation, namely adaptive background color estimation and motion detection. Subsequent clustering extracts regions of interest. The detections are then classified using color histogram features, which is detailed in Sect. 4. Finally, we project recognized player positions onto the ground plane and follow them over time. Each track is stabilized by a Kalman filter containing a physical motion model.

In this section, the image processing pipeline illustrated in Fig. 1 is presented. Mpeg attachment 1 shows a demo sequence starting with the panorama, visualizing the segmentation cues followed by classified clusters (colored ROIs) and the temporal integration (paths).

3.1 Full-HD panoramic video capturing

The image acquisition is realized by two stationary full-HD cameras (Prosilica GE1910C) with a color-CCD resolution of 1920 × 1080 pixels and a horizontal angle of view of 60°, each covering half of the field. The Gigabit-Ethernet cameras are directly connected to the processing machine, which is a four-core *Intel Xeon W3520* PC equipped with a *CUDA*[4] capable graphics card (*Geforce GTX 480*). For a computationally efficient and reliable backup parallel to image processing, the captured video streams are stored in raw format on a RAID-controlled storage ($\simeq$100 MB/s).

Basis of all later processing steps is a panoramic image composed from the two input images (cf. Fig. 2). Moreover, this feed has a higher usability for match reviews and other media purposes than a split view. For a proper mapping of image positions into field coordinates, the cameras and their poses are calibrated in advance. Compensation of radial lens distortion is applied within the stitching process and allows for a linear mapping (homography) from image to world coordinates [38]. The homographies are estimated from at least four-point correspondences per camera. These calibration points are chosen manually after installation by clicking, for example, the corner points and the end points of the center line within the distortion corrected image. The *Direct Linear Transformation* algorithm [15, Sect. 4.1] is used to compute the transformations for the two poses $\mathrm{H}_{1,2}$, which allow us to project image points from both camera coordinate systems (undistorted) to field coordinates.

By choosing an interpolated result pose $Hp = \alpha H_1 + (1 - \alpha) H_2$ for the panorama, we are able to map each of its positions x_p (pixels) back to a corresponding source position $\mathrm{x}_\mathrm{i} = \mathrm{H}_\mathrm{i}^{-1} \mathrm{H}_\mathrm{p} \mathrm{x}_\mathrm{p}$ either in the left, right, or both images (cf. Fig. 3).

Image data are recorded in *Bayer* format [2], which is converted to RGB employing an edge-adaptive, constant-hue demosaicking approach to avoid color corruption [14, 26]. To obtain color information from each pixel x_i we perform bilinear interpolation of the most proximate source pixels. For the area where the cameras overlap we interpolate between both gradually. As transformation parameters are assumed to be constant during a match the resulting image can be computed using a lookup table and parallel programming on the graphics card.

[4] Compute Unified Device Architecture by *Nvidia*, see http://www.nvidia.com/cuda.

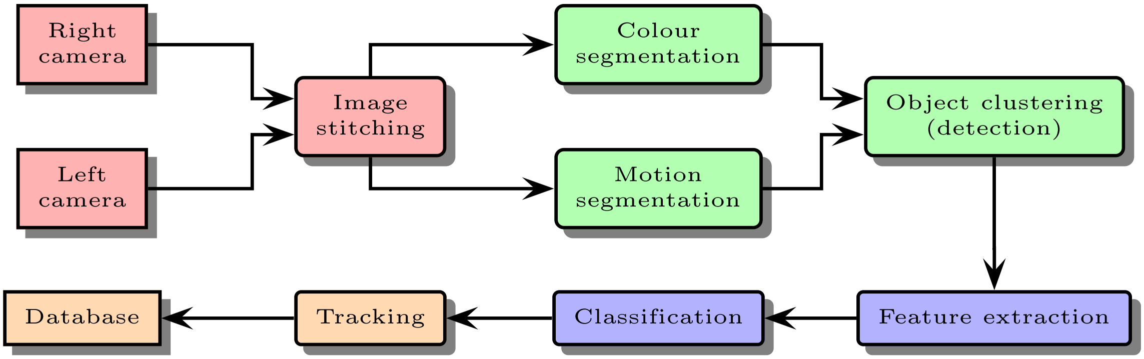
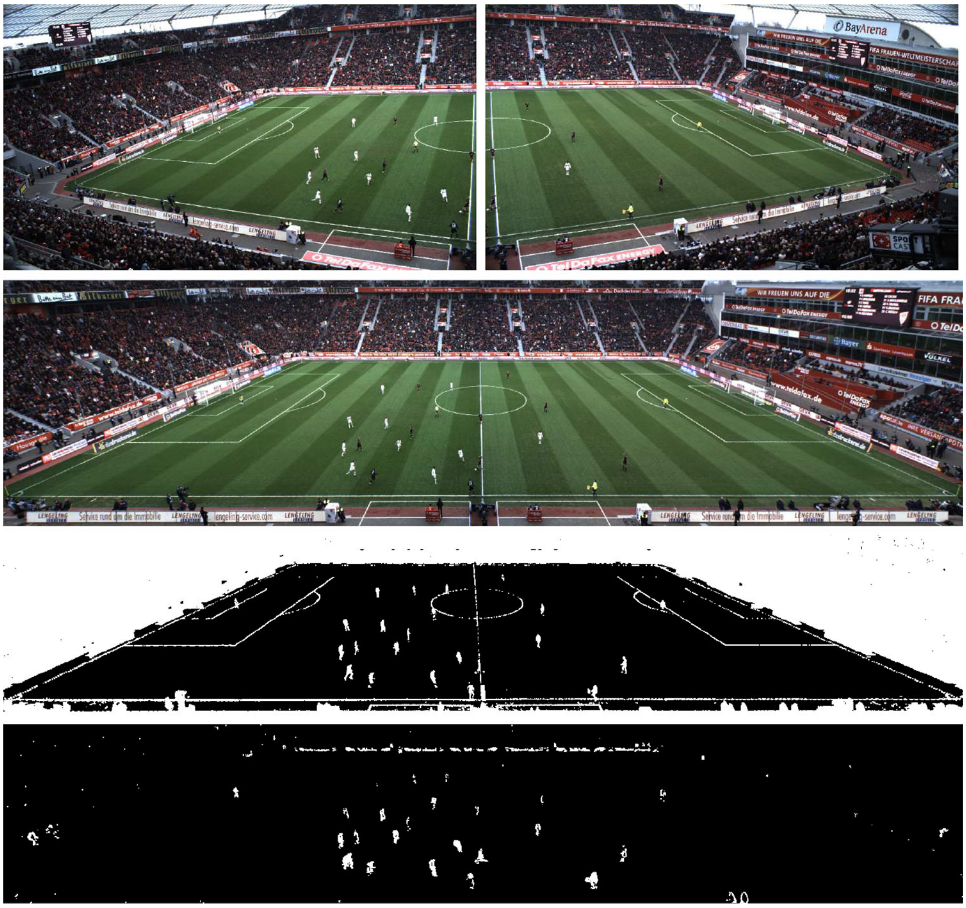


Fig. 1 System chart. *Colors* indicate independent module topics covered in Sect. 3

Fig. 2 Input images, panorama image, and segmentation cues (*color motion*)



3.2 Real-time object segmentation

Regions of interest (ROIs) are separated from the scene background performing two steps. We first extract pixel-wise segmentation cues (see Fig. 2) and then cluster conspicuous pixels locally to ROIs.

The color segmentation cue makes use of the plain-colored surface (i.e., grass-green), which is modeled by a multivariate normal distribution in HSV space. Therefore, all pixels covering the field area are taken into account. To remove outliers covering the background (i.e., players or line markings) we discard data exceeding a certain Mahalanobis distance, which is the distance to the center of the distribution measured in standard deviations. This is repeated on the inliers to improve the background distribution estimate. The estimated distribution is used to generate a lookup table assigning colors to background or foreground. Although the color estimate can be updated in regular intervals to deal with lighting changes, it will not help segmenting foreground into different-colored areas,

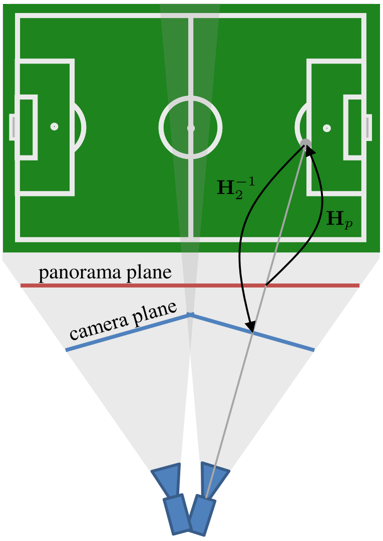


Fig. 3 Image planes and their corresponding transformations

for instance at lines, or in front of the perimeter boards/stands.

Therefore, a second cue for the detection of short-term color changes (“motion”) was considered. The background color b is modeled in each pixel individually by *exponential smoothing* with parameter α_b. The background color evolves over time from the pixel’ color p into $b_{t+1} = \alpha_b p_t + (1 - \alpha_b) b_t$. Given this background image, foreground pixels are identified by thresholding the current image subtracted from the background. This background model allows for motion perception under varying lighting conditions as the model adapts quickly to a new background characteristic (cf. [36]). In addition, we introduced a second parameter $\alpha_f < \alpha_b$ which is used if the new pixel value lies within the threshold distance (in color space) and is, thus, regarded as background. Otherwise α_f is applied, so that foreground measurements do not affect the background estimate much but still allow to adapt towards a persistent change in background on a longer time scale.

To increase robustness against noise, both cues are followed by morphological operators (cf. [10]). To this point, all operations take <10 ms for a HD panorama using an efficient GPU implementation (cf. Fig. 1, red and green modules).

The clustering algorithm is a region growing along “activated” cue-pixels, taking into account problem-specific knowledge (e.g., position of line markings, minimal or maximal object size).

Limiting the size of clusters to reasonable player dimensions (with some tolerance) has helped to improve robustness. Those limits are automatically determined from the world coordinates x_w of the segmented object (i.e., the player’s foot position), the camera position, a predefined interval of possible human heights (and widths), and the perspective transformation H_p. The vertical view angle of the camera is considered to determine the perspective shadow point x_s behind the player’s head. The height is estimated as the distance (in image rows) of mapped head and foot position $|(H_p^{-1} x_s - H_p^{-1} x_w)_y|$. The width is derived by regarding the object’s expansion Δ (orthogonal to the line of sight) within the ground plane. We transform the extreme points onto the image plane and receive a width estimate in pixels from $|(H_p^{-1}(x_w + \frac{\Delta}{2}) - H_p^{-1}(x_w - \frac{\Delta}{2}))_x|$.

Single ROIs may contain several objects overlapping each other. This matter is addressed by the subsequent classification module.

3.3 Multi-category classification

The development of the classifier is mainly driven by two issues—performance and time. As team clothing, background color and visual appearance vary to a large extent from match to match, we train the classifier during the preparation phase or at the beginning of a match. Therefore, classifiers which can be trained quickly and do not require too many training examples are necessary.

In addition to the obvious five classes: *outfielder 1/2*, *goalkeeper 1/2* and *referees*, we introduce an *error* category for irrelevant objects and a *group* class which applies to ROIs containing at least one outfielder from each team (see below). Preliminary experiments showed that a fully unsupervised learning approach (e.g., clustering) does not perform satisfactory. The procedures of feature extraction and classification for this special “online” learning task are detailed and evaluated in Sect. 4.

3.4 Multi-object tracking

Once all detected objects of a single frame are classified, they need to be matched to previously recognized ROIs to collect path data for each player. Therefore, ROIs’ root points are transformed to world coordinates and integrated over time within *tracks* (see Fig. 4). Each track is represented by a linear Kalman Filter [11, 36], which in contrast to that conventional time series filters support an explicit separation of the system dynamics (physical player model) and the process of measurement (positive classification at position z). The state is modeled as player position and velocity transitioning by laws of motion.

The Kalman process applied to the given task can be outlined by the following initialization: the matched root point in world coordinates defines the observation vector $z_t = [x_t \quad y_t]^T$ at time t. Both are uncertain observations of the state $x_t = [x_t \quad y_t \quad x'_t \quad y'_t]^T$.

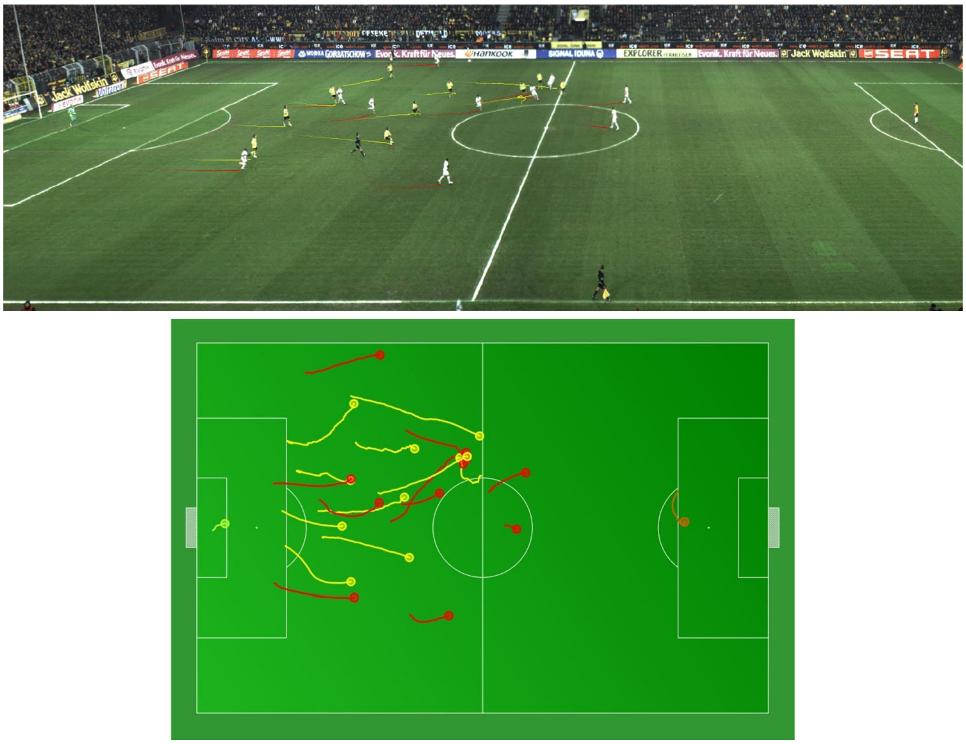


Fig. 4 Tracked paths for each player visualized in image and world coordinates

$z_t = H_t x_t + v_t$, where $H_t = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$ is the observation model mapping state space to observed space. Observation noise $v_t \sim \mathcal{N}(0, R)$ is modeled as a zero-mean Gaussian noise with covariance R.

One could incorporate the perspective transformation into the filter and define the observation noise in image coordinates. As this cannot be modeled in a linear Kalman filter, an *Extended* or *Unscented* Kalman filter [20] would be required. To ensure real-time capability, we simply assumed the noise for measuring the depth to be higher than for the lateral component: $R = \begin{bmatrix} \frac{1}{4} & 0 \\ 0 & 1 \end{bmatrix}$. Given the internal state at $t-1$ the filter dynamics assume the following true state x_t to emerge according to $x_t = F_t x_t - 1 + w_t$, where the state transition model

$$F_t = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

describes the physical behavior of a player in the ground plane and process noise $w_t \sim \mathcal{N}(0, Q)$. Process noise is basically introduced by the non-modeled acceleration with variance σ_a^2, which translates to

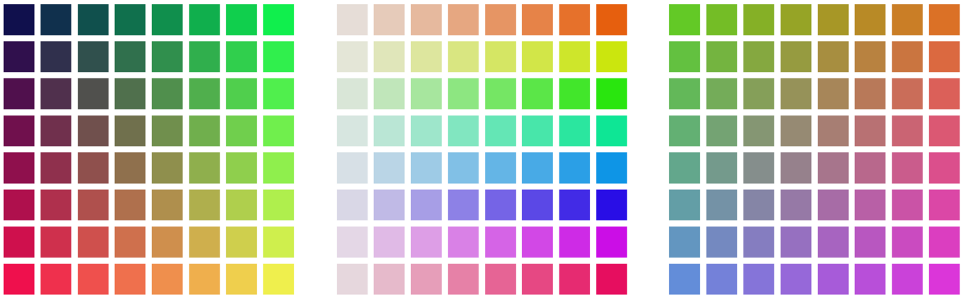
$$Q = \begin{bmatrix} \frac{\Delta t^2}{2} \\ \Delta t \end{bmatrix} \begin{bmatrix} \frac{\Delta t^2}{2} \\ \Delta t \end{bmatrix}^{\mathrm{T}} \sigma_{\mathrm{a}}^2 = \begin{bmatrix} \frac{\Delta t^4}{4} & \frac{\Delta t^3}{2} \\ \frac{\Delta t^3}{2} & \Delta t^2 \end{bmatrix} \sigma_{\mathrm{a}}^2$$

where σ_{a} should be chosen in a physically reasonable range (here: $\sigma_{\mathrm{a}} = 3\,\frac{m}{s^2}$). The presented filter model is able to robustly estimate the player's trajectory and to predict his/her position in the next time step in real-time.

Consequently, we are left with a constrained matching problem of new measurements and the Kalman predictions in each frame. *Groups* are treated as "jokers" that are able to serve as an update for more than one track of different classes. As we do not distinguish between players of the same team, a human operator needs to assign "new" tracks to individual players to complete the database. Experiments show sufficient tracking performance for an operator to easily assign all players of one team in real-time (cf. Sect. 5.6).

4 Classification

This section focuses on the classification module. We present image features and classifiers considered in the following empirical evaluation.



(a) R and G; B = 0.3 (b) H and S; V = 0.9 (c) U and V; Y = 0.5

Fig. 5 Visualization of two-dimensional color histograms with a resolution of three bits (i.e. eight bins) in each dimension (row, column)

4.1 Feature extraction

The requirements for valuable image features in the given scenario are

- low dimensionality,
- fast computability, and
- good class discrimination.

Due to the fact that we detect non-rigid objects, recognition should be invariant to player pose and orientation. Moreover, team shirts are designed to be distinguished well by their color. Therefore, we decided not to use shape or spatial information but color histograms, which proved to be valid features for object classification [6, 39].

The images are recorded in RGB space but there are good reasons to evaluate other color spaces. An HSV representation separates the color properties hue, saturation, and brightness, and enables us to rely less on those ones the recognition process should be invariant against. The YUV space has a single brightness channel and defines the hue in two dimensions without periodicity. Full resolution three-dimensional histograms result in feature vectors of size 256^3. Discarding channels and/or reducing their resolution makes the histograms usable in the given scenario and generally less prone to noise (cf. Fig. 5). Histogram entries are normalized to cope with varying size and aspect ratio of the detected image regions.

Given the foreground segmentation (see Sect. 3.2) we are able to identify object relevant pixels in each detected image region. Thus, only those pixels are considered for the histogram. Preliminary experiments showed a significant increase in performance using this more descriptive representation.

As a benchmark for feature extraction, we considered two state-of-the-art methods in the given context. We regarded image features extracted by PCA applied to uniformly scaled RGB training images (see Fig. 6). principal component analysis is arguably the best-known linear feature extractor [18]. It was successfully employed for several recognition tasks based on the *Eigenface* approach [32] and lately in the context of sports analysis to distinguish between players' body postures [21].

As a second reference method, *spatiograms* proposed by [5] were applied. *Spatiograms* extend color histograms by spatial information without the need of preset image regions. Each histogram bin additionally stores the mean position and covariance matrix of its associated pixels and, thus, enables a classifier to learn spatial relationships during training.

4.2 Real-time classification

We evaluate different real-time capable classification algorithms: linear discriminant analysis (LDA), nearest neighbor (NN), and one-vs-all multi-class support vector machines (SVMs).

The time spent on training these classifiers as well as the time they need for classification crucially depend on the features used and the size of the training set. For our experiments we only considered combinations that are real-time capable in the given setup. The model selection is realized by grid search and cross-validation, independently for each dataset. Runtimes reported in the experimental evaluation always include the time needed for model selection.

Linear discrimination using LDA gives surprisingly good results in practice despite its simplicity. Dealing with underrepresented classes, we apply regularization to ensure proper conditioning of the covariance matrix in LDA [16, Sect. 4.3.1].

Nearest neighbor classifiers are of particular interest due to their fast training. We employ class-wise hierarchical clustering of training examples to reduce the amount of prototypes and, thus, guarantee real-time classification [16, Sect. 13.2.1], [18]. The distance of two training examples is defined by their Euclidean distance in feature space. Clustering is performed in an agglomerative complete-linkage fashion, separately for each class, until the desired number of prototypes is reached. Finally, the classification decision for test data is given by the class label of the nearest cluster (i.e., 1-NN).

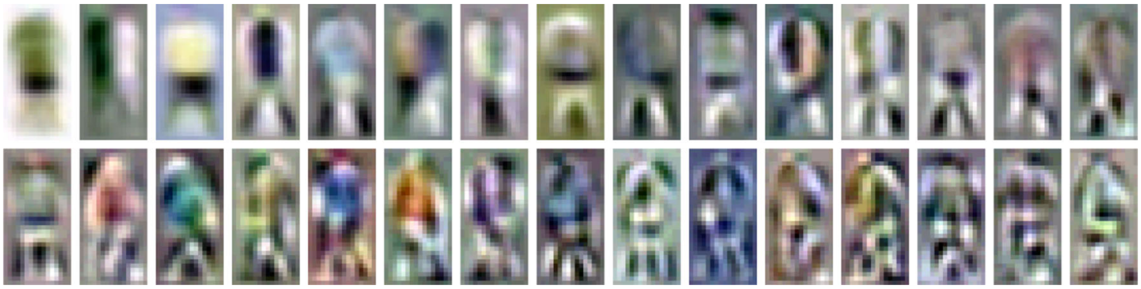


Fig. 6 First 30 principal components of dataset I (cf. Fig. 7, *top row*). Contrast and saturation adjusted for visualization

Support vector machines [7] mark the state-of-the-art in binary classification. They are theoretically well-founded and usually show excellent classification results in practice. However, the training time of non-linear SVMs scales unfavorable with the number of training patterns. There are multiple extensions of SVMs to multi-category classification. In this study, we consider the popular one-versus-all approach [28, 34]. For fast training of the SVMs, we use the optimization algorithm proposed by Glasmachers and [17].

4.3 Evaluation

For the evaluation of the classification module the following questions were addressed:

- Which features/classifiers can we use at all in the given scenario?
- Are the proposed color histograms (see Sect. 4.1) powerful features for player recognition?
- Is there a common best feature setup (i.e., color space and histogram resolution) for our application?
- How does the classification performance scale with collection/training time?
- How long does it take from the beginning of the data acquisition until we have a reliable classifier?
- Does our solution offer sufficient performance, in particular, for the underrepresented classes?

4.3.1 Setup

For evaluation, three datasets covering matches from different stadiums and various team clothing were collected (see Fig. 7). The image data were extracted by the segmentation algorithm detailed in Sect. 3.2 followed by the considered feature extraction. Data were sampled with a frequency of 1 Hz. To mimic the *live* collection and classification task, training sets only contain images from the first couple of minutes of each match (including the running-in period) and the test data are drawn from the rest of the game. Therefore, they are not independent and identically distributed (i.i.d.). We do not address this problem explicitly (this is a direction for future work). Each training and test dataset contained about 4,000 and 2,000 examples, respectively.

The SVMs and LDA are based on the implementations in the Shark[5] machine learning library [17]. The SVM model parameters were selected through grid search from

$$C \in \{1, 10, 100, \ldots, 10^4\} \text{ and}$$

$$\gamma \in \{2, 3, 4, \ldots, 25\} \text{ with kernel}$$

$$k(x, z) = \exp(-\gamma \|x - z\|^2).$$

The LDA was regularized by adding

$$\hat{\sigma}^2 \in \{10^{-4}, 10^{-3}, \ldots, 1\}$$

to the diagonal elements of the empirical covariance matrix. Regularizing LDA can lead to better generalization [16] and ensures numerical stability at the same time. The NN operated on up to 50 clustered prototypes per class. All parameters were determined by threefold cross-validation independently for each dataset.

We conducted experiments for all valid pairs of features and classifiers to identify the best-performing combinations. As mentioned in Sect. 4.1, we compared the proposed features with PCA and spatiograms. We extracted about 80 principal components, which explain 90 % of the variance (computed as sum of used eigenvalues by the sum of all eigenvalues of the data covariance matrix). The spatiograms were applied as proposed by Birchfield and Rangarajan [5], which basically results in a YUV histogram with a bit resolution of (2:3:3) extended by mean and variances.

4.3.2 Results

Looking at the influence of the chosen features, Table 1 documents the overall and individual performance on all datasets with maximal number of training examples. The results identify setups that violate time constraints either for training (<2 min) or for test (<20 ms for 40 examples

[5] See http://shark-project.sourceforge.net

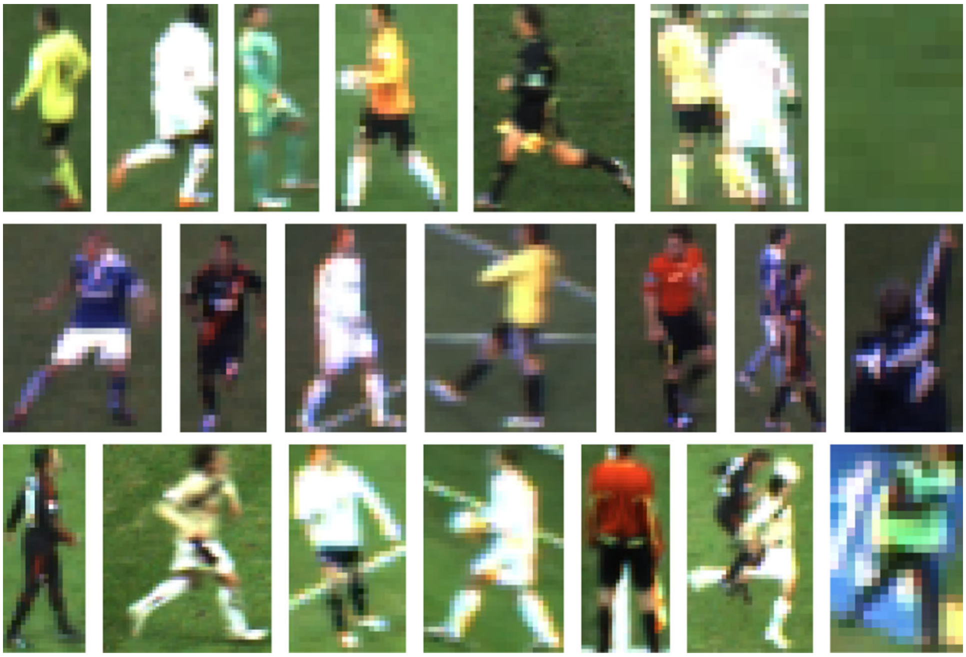


Fig. 7 Examples from the three datasets (*one per row*). *Left to right*: outfielder team 1/2, goalkeeper 1/2, referees, group, error

per frame). Throughout all experiments, we found superior performance of the color histograms. Moreover, PCA is much more expensive as it has a complex training phase, needs image scaling and has a rather high-dimensional input space. We assume that the rigid spatial mapping of PCA features impairs performance in many situations, for example, in the case of inaccurate segmentations or strongly varying player poses. Learning poses explicitly might require more data.

Similar conclusions can be drawn from the spatiogram experiments. The strong increase in training time is caused by the feature vector expansion (by a factor of 5). Results show that even the very flexible spatial information offered by spatiograms do not improve performance.

We were able to identify a channel resolution for each of the two color spaces, HSV and YUV, that is superior to the rest. It is noteworthy that these solutions have a lower resolution of the brightness channel (V and Y, respectively), supporting robustness towards varying lighting conditions. Nevertheless, discarding the brightness decreases performance.

Moreover, the one-versus-all SVM approach outperformed alternative methods in terms of accuracy. In average the SVM yielded a 50 % lower classification error compared to LDA.

Keeping in mind the need for a classifier that offers proper results early in the match, we examined the performances of each method after certain periods of time. To receive a meaningful estimate in the given context, we took into account the time spent for data collection, annotation, and training. Figure 8 illustrates this using the example of dataset I.

The NN was fastest and yielded the best results within the first minute. In the beginning, the non-i.i.d. issue discussed above is most severe and presumably 1-NN can cope better with it than the other classifiers. The best-performing SVM needed several minutes which is a drawback in practice. A reasonable trade-off solution was to employ NN classification at the beginning of each match—waiting for the SVM to take over. In this case, we are moreover able to support the data collection phase by presorting the training examples with the NN class labels. This reduces manual annotation effort compared to the use of clustered data and, thus, allows for an earlier SVM training (cf. Sect. 5).

Figure 9 presents the class-wise precision and recall, which document the SVM's high performance even for the underrepresented classes. In particular, the recall percentage was close to 100 % throughout the six relevant classes. That is, once a relevant player is detected, it is then classified correctly, which is desirable for the application at hand. Minor precision for the smaller classes was caused by misclassification of images from the error-class. We are able to compensate this using temporal integration within the subsequent tracking module.

4.3.3 Final evaluation

For a final validation of the documented results, six further datasets were recorded. To evaluate the stability of the chosen SVM-based classifier throughout an entire match, examples were drawn from ten equidistant intervals of 2 min. Thus, each match generated ten sets containing ≈5.000 samples each.

Table 1 Overall error rate (0/1 loss) for the best-performing classifier of each method, i.e., the one-versus-all SVM and the NN using 50 prototypes per class

Col. res. [bits]:		RGB		HSV							YUV				PCA	Spatiograms
Dataset	Method	2:2:2	3:3:3	6:0:0	3:3:0	4:4:0	2:2:2	3:3:2	3:3:3	4:3:2	0:4:4	2:2:2	2:3:3	3:3:3		
I	LDA	7.3	6.3	8.9	8.7	9.0	10.2	7.2	6.7	7.3	9.0	10.3	6.5	5.9	9.5	7.6
	SVM	4.7	–	5.3	4.0	6.9	6.2	2.9	–	–	–	5.3	3.3	–	8.3	10.3
	NN	4.1	6.5	6.5	6.3	7.3	7.7	5.2	5.1	4.5	4.8	6.0	2.8	6.0	18.2	5.8
II	LDA	4.8	3.1	13.2	9.6	9.2	3.8	2.0	2.3	2.3	13.1	11.3	8.8	7.5	13.9	3.4
	SVM	1.5	–	6.7	4.2	4.0	1.7	1.8	–	–	–	3.1	2.7	–	11.7	9.2
	NN	3.2	1.3	10.1	6.7	5.9	3.6	2.0	2.1	2.1	6.2	4.4	3.9	3.2	19.5	12.4
III	LDA	16.8	10.3	19.9	23.2	18.4	11.4	10.5	9.5	9.3	22.3	23.6	11.6	10.1	9.2	7.2
	SVM	6.6	–	8.2	9.1	8.3	4.6	4.2	–	–	–	5.9	4.6	–	5.1	14.2
	NN	13.2	6.5	17.3	16.5	10.4	9.2	7.7	6.8	7.0	24.9	15.7	13.2	9.0	16.9	18.8
Training time (s)	LDA	1	81	1	1	11	1	11	79	82	11	1	11	73	273	1.3×10^6
	SVM	30	133	39	31	94	26	70	143	146	150	54	66	134	442	1.4×10^6
	NN	6	27	6	6	15	6	14	26	27	14	6	14	25	274	1.2×10^4
Classification per ex. (ms)	LDA	0.03	2.10	0.03	0.03	0.50	0.03	0.50	2.10	2.10	0.49	0.03	0.49	2.10	0.96	14.51
	SVM	0.03	0.19	0.05	0.04	0.14	0.03	0.13	0.24	0.27	0.10	0.04	0.09	0.19	1.81	5.38
	NN	0.05	0.38	0.05	0.05	0.20	0.05	0.19	0.38	0.38	0.19	0.05	0.19	0.38	0.92	0.95

The last six rows indicate the computational complexity (runtimes) of feature/classifier combinations for training and classification of a single image patch

Figure 10 depicts the mean performance of the two examined configurations over time. The predominance of feature configuration *HSV 3:3:2* was confirmed (cf. Table 1). A mean classification rate of about 98 % was reached. The standard deviation of all subsets was 1.8 %, the minimum performance 90.5 %. Notably, the validation data included a team with green shirts, which did neither impair classification nor segmentation performance.

5 Human–machine interaction

By a considerate choice of features and machine learning algorithms, our classification module achieves very high accuracy in object recognition. Still, the multi-object tracking problem at hand, namely distinguishing between similar dressed players in crowded scenes, can hardly be solved fully automatically over the period of a whole match.

The aspired classification performance can only be achieved by supervised learning methods (see Sect. 4) that have to be trained to perform well under local conditions (e.g., lighting and shirt colors). Therefore, we need to systematically integrate a necessary number of human operators, the recognition system, and the consumer access into one efficient technical framework.

In this section, we propose operator integration methods that feature mobility, high data accuracy, and low personnel expense through a powerful interface to the distributed system sketched in Fig. 11.

5.1 System setup and calibration

Transportation and setup of the hardware can be easily managed by two operators. Figure 12 shows our setup at two different sites. As introduced in Sect. 3, the generation of the panorama video and later tracking requires an image-to-world mapping which is derived by manual registration of at least four points in each camera image. At this stage, as all external parameters are known, the fully automatic calibration of the background segmentation algorithm follows.

After establishing a local database connection we prepare an entry for the upcoming match to be referenced in all tracking datasets generated later. With *SQL* the database has a well-defined interface and represents the central data storage, shared within the distributed system.

With the beginning of the match, two remote operators control the system via laptops. With only little space in the stadium—especially at the camera stand—a remote architecture with only two regular seats on the press gallery is a very feasible setup.

5.2 Human supervision of machine learning

Data collection and annotation are naturally manual processes. Nevertheless, we drastically reduce human effort to make supervised learning techniques applicable in the given scenario.

Figure 13 illustrates the situation shortly before the beginning of a soccer match. Teams enter the stadium, line

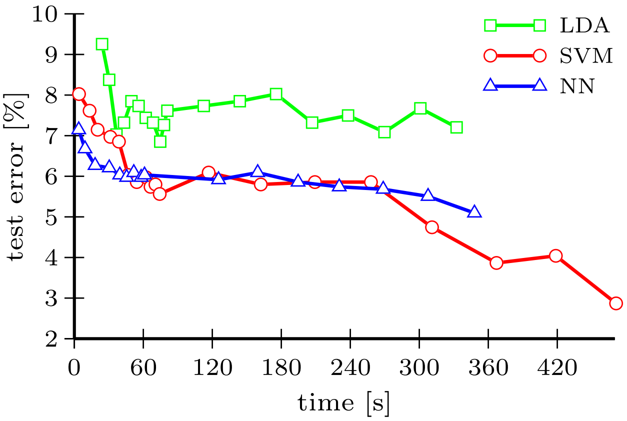


Fig. 8 Behavior of test error in relation to data collection and training time from dataset I

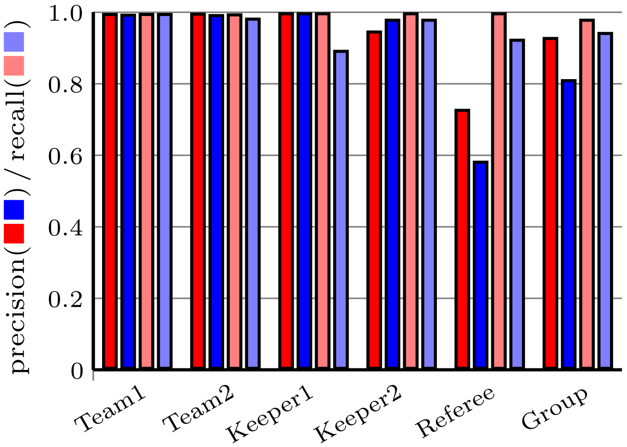


Fig. 9 Precision and recall per class for the final SVM (*red*) and NN (*blue*)

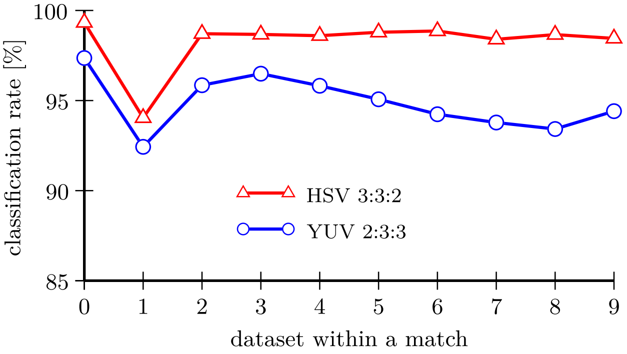


Fig. 10 Validation of SVM classification showing the mean performance throughout six further matches. Thereby, each test set $n = \{1, \ldots, 9\}$ was classified by an SVM trained on examples from sets $0, \ldots, n - 1$. Thus, index 0 is a training error

up, and finally distribute for the kick-off. The whole procedure normally takes 2 or 3 min. This is generally the first time that the team shirts are visible (during warm-up usually different shirts are worn) and, therefore, the first opportunity to collect valid training examples.

As it is required to gather all statistics from the very beginning of the match, a first classifier has to finish its training during this short period. To address this requirement and with the knowledge of the preceding sections, we propose a combination of *unsupervised* and *supervised* learning in an *iterative* approach.

As soon as the teams enter the stadium, ROIs are collected and a *clustering* algorithm (i.e., *unsupervised learning*), based on the same color histograms used for classification, is employed. The resulting clusters of similar images are instantly presented to the human operators (see Fig. 14) who pick a couple of representative examples for each of the seven classes if possible (cf. Fig. 7 in Sect. 4.3). Shortly before the match starts, we create an 1-NN classifier based on that early data. Thus, tracking and data collection start off.

This early classifier cannot work perfectly robust for several reasons. For instance, it is not guaranteed to collect sufficient examples of groups or misdetections in the first minutes. The NN classifier will then assign such images to any of the other classes. Apart from these issues, as stated earlier in Sect. 4.3, the classification accuracy can generally be increased using a more sophisticated classifier incorporating more training examples and more training time.

Therefore, the collection phase is continued in parallel, but at this stage enhanced by taking into account the NN class assignments for automatic pre-sorting. The resulting clusters are presented to the human operators. Although they are already engaged in resolving tracking conflicts and identification of players (see below), they can still approve the results of unsupervised data collection during game interruptions.

The SVM training starts as soon as enough examples are available for each class. This number is empirically chosen and increases with the number of classes and the dimensionality of the feature vectors. Due to *cross-validation* for model and feature selection (cf. Sect. 4), the training itself is completely automated and does not need human input.

5.3 Identification of individual players

The classifier is able to distinguish different clothes but does not identify each player in person. The shirt numbers that could be used for this purpose are not reliably recognizable even using HD cameras. It is noteworthy that skilled humans are able to identify players in the videos, incorporating different hints like players' physique, skin

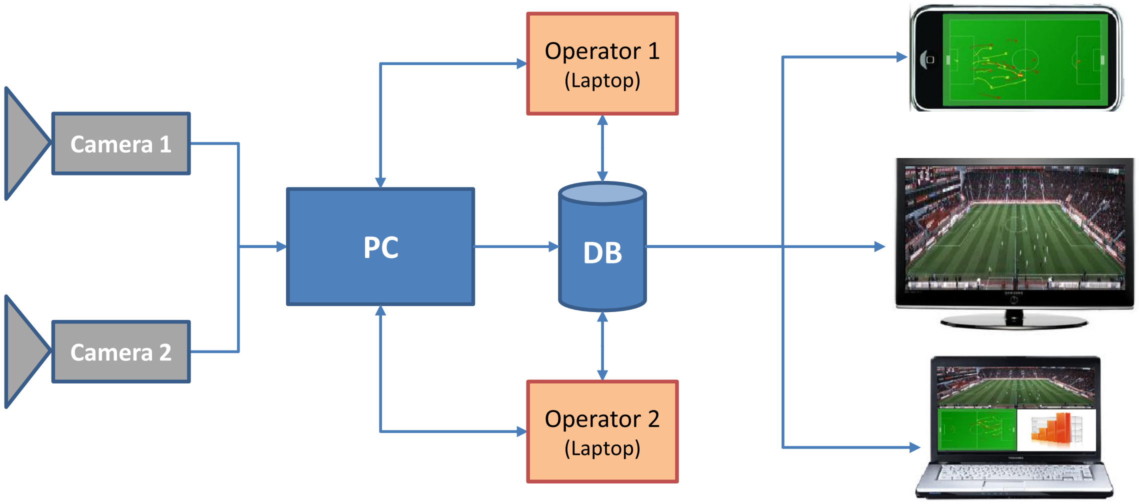
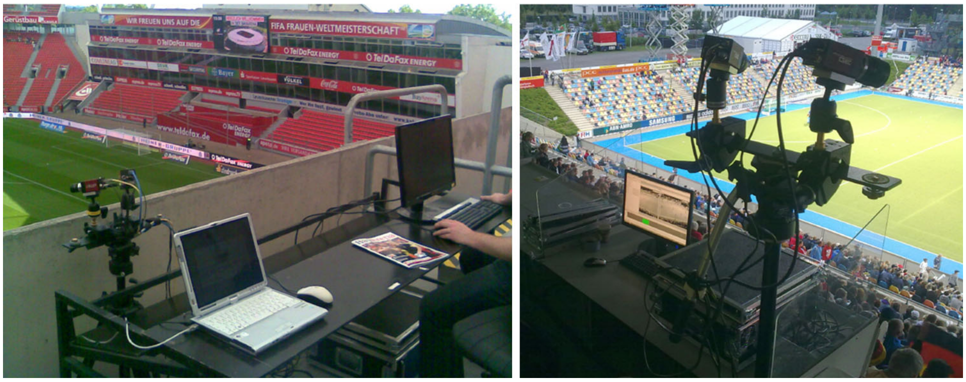


Fig. 11 Human–machine interaction overview: operators' and consumers' interface

Fig. 12 System setup for live acquisition in a German soccer stadium (*left*) and at an international field hockey tournament (*right*)



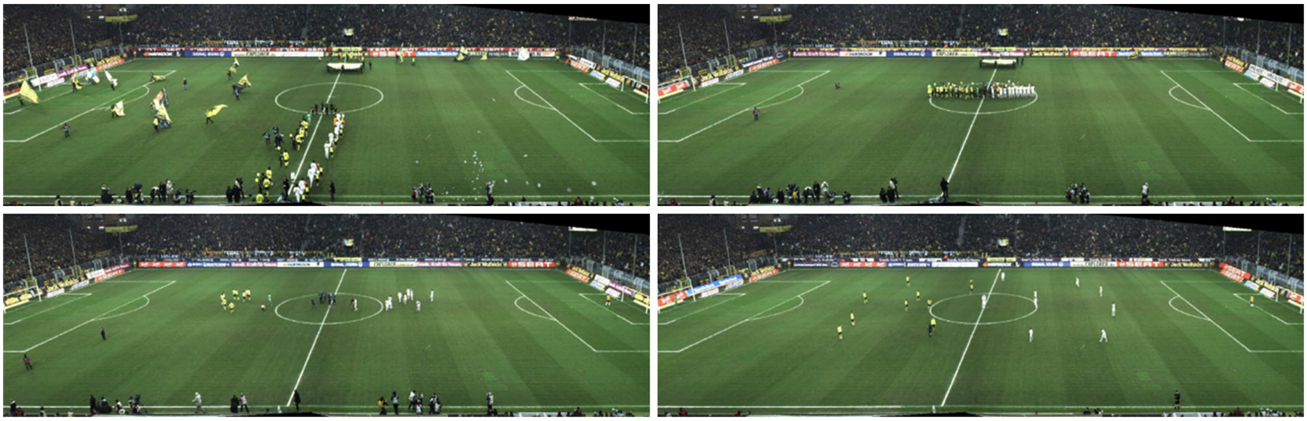


Fig. 13 Running-in of the teams. Typical activities during the last 2 or 3 min before the start of a match

and hair color, course of motions, and position relative to the team.

We propose the integration of two operators—one for each team—to assign player identities and enhance recognition performance. The operators are supplied with all recognized player tracks, visually integrated into the live panorama video feed. As all tracking data are written to the database in real-time, the operator only needs to add the players' personal tag in order to complete his dataset. Starting the match, all tracks are still anonymous and

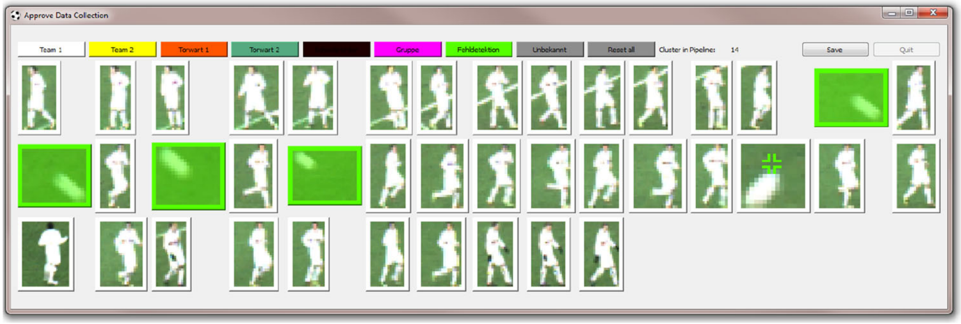


Fig. 14 Operator interface for the assignment of pre-sorted object clusters to the trained classes. The shown cluster is assigned to one of the classes indicated by the *colored buttons* in the *top row*. Examples that do not fit this class are labeled individually. By confirming the dialog, the assignments are sent to the processing machine

require identification. Tracked players are marked by a colored bounding box coding their status. Tracks that require operators' action are highlighted by signal colors while identified tracks carry their corresponding shirt number (see Fig. 15).

The assignment is solved through an intuitive *one-click* interface visualized in Fig. 15. The *mouse menu* shows shirt numbers, arranged according to the team's tactical formation. With the given interface, the operator is able to assign identities to all players of *his* team and keep track of them while following the match. As tracks typically last several minutes, this does not cause too much workload.

Rarely it may occur that operators are not able to identify all tracks in real-time. In this case, open tasks are queued for later treatment (e.g., during oncoming interruptions). This task list is integrated in the operators' main GUI allowing short video replays for each task (see Fig. 16). For superior visibility of the field and individual players, the operator's view is flexible supporting zooming and scrolling within the full panorama.

5.4 Solving recognition conflicts

Although the classification of single images works robustly with the presented approach, multi-target tracking does not perform sufficiently reliable in every situation. All team sports comprise scenes in which even the most sophisticated tracking approaches run into problems:

- Players wearing same shirts occlude each other for a longer time.
- Corners or free-kicks result in crowded areas leading to multiple occlusions and even incomplete player segmentation.
- Players leaving and entering the field for treatment or due to exchange.
- Non-relevant individuals enter the field (e.g., medics, fans).

Thus, none of the currently published player tracking systems is able to generate valid statistical data over a whole match without continuous supervision.

Given our *Kalman filter*-based multi-object tracking approach (see Sect. 3.4) confidences for each track are derived. In particular, we are able to identify uncertain situations where we cannot guarantee valid data. In this case, the operators' attention is drawn to the conflict requiring his approval. One special case of conflicting tracks arises from two or more players from the same team approaching and possibly taking over each others' tracks. This uncertain situation is indicated by the bounding boxes mentioned above, allowing for an instant identity swap.

5.5 Consumer access and data visualization

The information stored in the database can be used instantly for a variety of applications: fans in the stadium directly access detailed statistics using smartphones, the media improve live TV coverage with interesting facts, and finally coaches are offered a valuable tool for detailed analysis already during the match.

Having those applications in mind, we process the raw data to provide different flavors of information visualization—exceeding conventional quantitative data plots and tables (see Figs. 17, 18). Tracking data of individual players, for example, can either be visualized in an artificial bird's eye view or integrated into the camera perspective (cf. Fig. 4).

Apart from collecting statistics online, a full high-definition Mpeg video of the panorama view is generated. This is available immediately after the match for detailed review together with the collected data.

5.6 Experimental HMI evaluation

During development all modules were regularly tested by different subjects with background in sports analysis. Thereby we recorded data and performed live experiments in seven different stadiums. The system has been extensively evaluated concerning the following aspects:

- *Field calibration* The calibration for generating the live panorama video was tested successfully on-site for different sports fields (soccer, hockey, tennis).

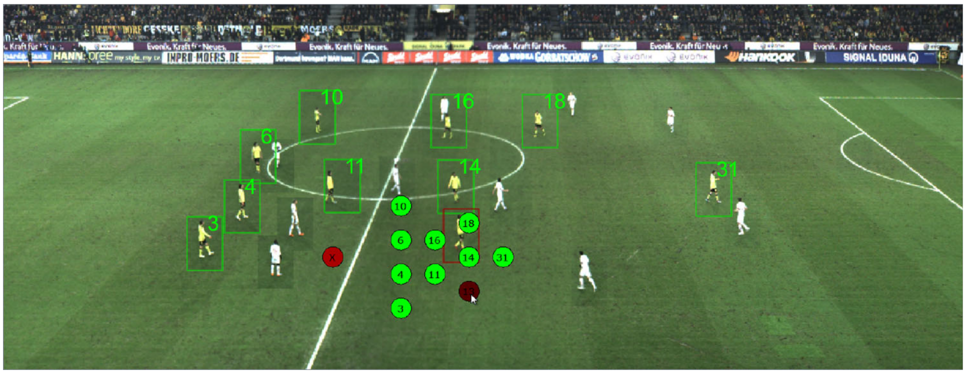


Fig. 15 Operators' mouse menu for *one-click* player assignment

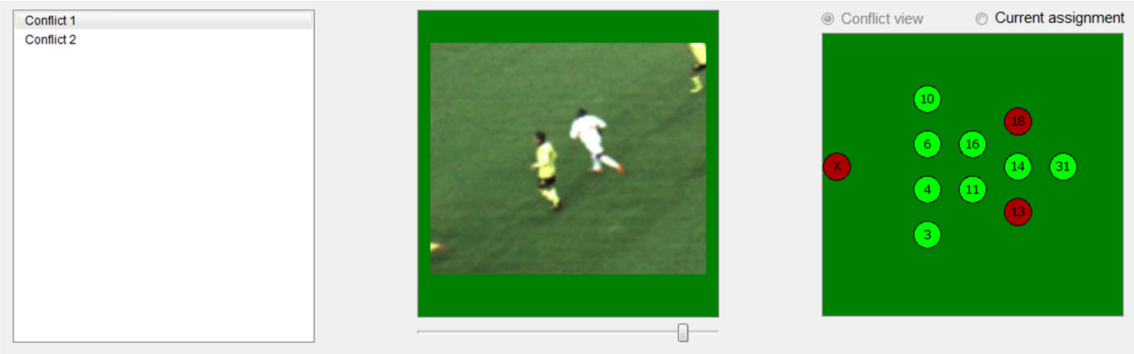
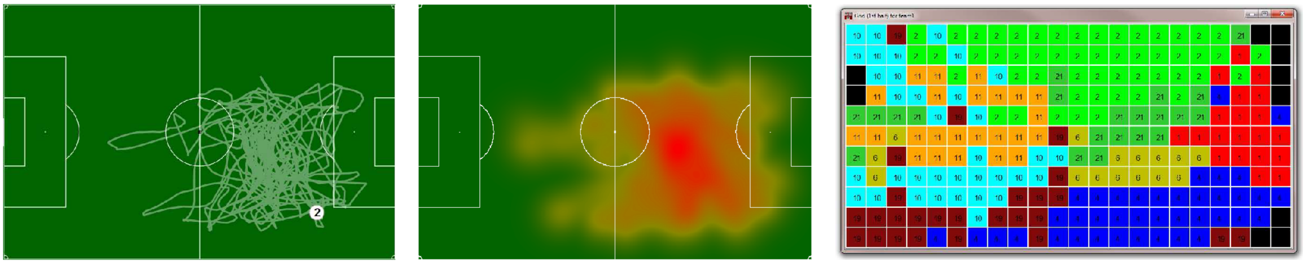


Fig. 16 Conflict review: the operator is presented a video clip of the missed situation and corresponding team assignments—again one click solves the problem



(a) Raw player path **(b)** Heat map indicating a spatial probability distribution **(c)** Team grid showing the most frequent player for each grid cell

Fig. 17 Offered positional data visualization

- *Segmentation* Automatic real-time player segmentation using an adaptive background model worked robustly. Experiments were conducted under various external conditions (i.e., weather, ground texture).
- *Accuracy test* Human subjects were tracked on predefined paths to evaluate the tracking accuracy against ground-truth data. The error of the measured distances did not exceed 3 %.
- *User feedback* Sports scientists have accompanied the development of useful data preparation techniques. Customers' feedback helped us to refine visualization methods and validated their relevance for professional sports analysis.
- *Full system test* The complete system, including operators and database, was deployed successfully during an official match of the highest German soccer league (Bundesliga) and assessed offline with several recordings.

These experiments were conducted to assess the commercial applicability of the system, however, not with a strict scientific protocol.

6 Conclusions

Video-based sports analysis is an active field of research. The resulting *live* data are valuable, among others, for

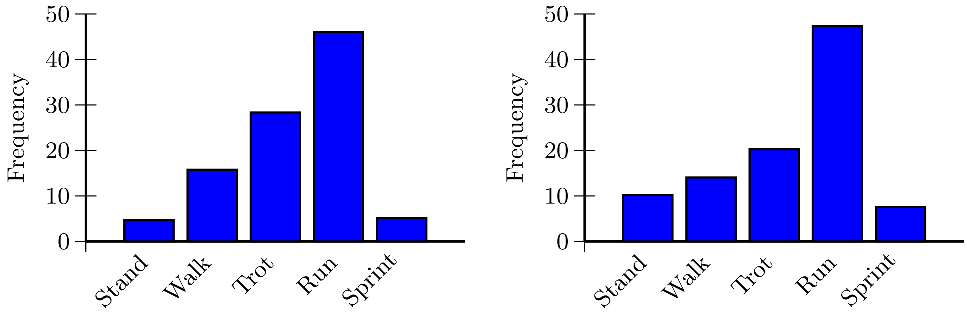


Fig. 18 Speed histograms of two players based on tracking data from a whole match

professional match analysis, media coverage, and sports betting. But also fans (at home or onsite in the stadium) are interested in more detailed statistical information.

This paper proposed a self-contained system for video-based sports analysis featuring high accuracy, mobility and low system cost. While recognition systems such as the one presented here can operate autonomously in many situations, human operators are still needed to assure high reliability needed for the applications mentioned above. The contribution of this study is twofold.

On the one hand, we pointed out the appropriate integration of human operators into the processing chain. Regarding the limitations of the given working environment (i.e., crowded stadium), we designed an efficient system architecture keeping interactions intuitive and as simple as possible. The system was successfully tested at official sports events. Valid results were collected *online*, providing positional data live through a slim database interface. The experiments proved low manpower requirement for the supervision of our recognition system. We showed that collection of individual player statistics in real-time is possible by incorporating two human operators only.

A second focus was put on the classification task. Due to the live scenario, this module is constrained in terms of time spent on data collection, training, and classification. Evaluating color histogram features together with either *nearest neighbor combined with clustering* (NN), *linear discriminant analysis*, or *support vector machines* (SVMs) we were able to achieve an overall misclassification rate of 1.8–4.2 % throughout different datasets, obtaining a close to 100 % recall for the six relevant classes.

Performance crucially depended on the choice of histogram resolution and less on the color space itself. The proposed histogram features outperformed state-of-the-art baselines, namely PCA (applied to RGB training images) and spatiograms, significantly. These findings support our hypothesis that spatial information is not necessary to solve the classification task at hand. Moreover, experiments showed superior performance of the NN approach for early classification (i.e., considering only examples from the first few images). After a longer collection phase (>5 min), SVMs outperformed alternative classifiers.

These findings suggest a two-stage solution, using NN as an *ad-hoc* classifier first, which is then replaced after some minutes by a fully trained SVM operating for the rest of the match. Further evaluation of the SVM-based classifier was carried out on a validation dataset taken from six matches. With a mean classification rate of 98 %, the high performance was confirmed.

To improve classification performance, it is intended to apply domain adaptation techniques to account for the difference in class distribution between training and test data (also known as class imbalance problem, see [19]). For SVMs in binary classification, a suitable approach was already proposed [22] and could be transferred to the multi-class problem. Moreover, we think it is worthwhile to look into new developments in semi-supervised learning for reducing labeling cost (e.g., *group induction* as proposed by [31]). We will speed up the training times of the employed classifiers even further by also utilizing the GPU (e.g., see [12] for an overview of efficient nearest neighbor classification on GPUs) and more efficient training and model selection algorithms available in the forthcoming Shark release [17].

A further research direction of our project is the analysis of the extracted statistical data by means of data mining to identify behavioral and tactical patterns of teams. Therefore, not only spatial but also temporal features need to be captured on different time scales. This will support the acquisition of higher level statistical data, so-called *event data* (such as an automatic indexing of corner, free-kick or even one-on-one situations), and an automatic scene categorization for systematic match reviews and error analysis. Looping this information back, the recognition process will profit from a learnt player distribution for each team to realize a fully automatic identification of players in person. This should be backed up by recognition of shirt numbers and additional visual cues.

Acknowledgments The authors would like to thank the German Federal Ministry of Economics and Technology for supporting their work through a funded project over a period of 2 years based on the basis of a decision by the German Bundestag (Grant KF2449001 MS9)

References

1. Assfalg, J., Bertini, M., Colombo, C., Bimbo, A.D., Nunziati, W.: Semantic annotation of soccer videos: automatic highlights identification. Comput. Vis. Image Underst. **92**(2–3), 285–305 (2003)
2. Bayer, B.E.: Color imaging array. Eastman Kodak Company. US Patent 3971065, 1975
3. Beetz, M., von Hoyningen-Huene, N., Kirchlechner, B., Gedikli, S., Siles, F., Durus, M., Lames, M.: ASpoGAMo: automated sports game analysis models. Int. J. Comput. Sci. Sport **8**(1), (2009)
4. Ben, Shitrit, H., Berclaz, J., Fleuret, F., Fua, P.: Tracking multiple objects under global appearance constraints. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV), pp. 137–144, (2011)
5. Birchfield, S.T., Rangarajan, S.: Spatiograms versus histograms for region-based tracking. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 1158–1163, (2005)
6. Chapelle, O., Haffner, P., Vapnik, V.: Support vector machines for histogram-based image classification. IEEE Trans. Neural Netw. **10**(5), 1055–1064 (1999)
7. Cortes, C., Vapnik, V.: Support-vector networks. Mach. learn. **20**(3), 273–297 (1995)
8. D'Orazio, T., Leo, M.: A review of vision-based systems for soccer video analysis. Pattern Recognit. **43**, 2911–2926 (2010)
9. D'Orazio, T., Leo, M., Spagnolo, P., Mazzeo, P.L., Mosca, N., Nitti, M., Distante, A.: An investigation into the feasibility of real-time soccer offside detection from a multiple camera system. IEEE Trans. Cir. Sys. Video Technol. **19**(12), 1804–1818 (2009)
10. Figueroa, P., Leite, N., Barros, R., Cohen, I., Medioni, G.: Tracking soccer players using the graph representation. In: Proceedings of the International Conference on Pattern Recognition vol. 4, pp. 787–790, (2004)
11. Gelb, A.: Applied Optimal Estimation, 1st edn. MIT Press (1974)
12. Gieseke, F., Heinermann, J., Oancea, C., Igel, C.: Buffer k-d trees: processing massive nearest neighbor queries on GPUs. In: Proceedings of the International Conference on Machine Learning (2014)
13. Glasmachers, T., Igel, C.: Maximum-gain working set selection for support vector machines. J. Mach. Learn. Res. **7**, 1437–1466 (2006)
14. Gunturk, B.K., Glotzbach, J., Altunbasak, Y., Schafer, R.W., Mersereau, R.M.: Demosaicking: color filter array interpolation. IEEE Signal Process. Mag. **22**(1), 44–54 (2005)
15. Hartley, R.I., Zisserman, A.: Multiple View Geometry in Computer Vision, 2nd edn. Cambridge University Press (2004)
16. Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer-Verlag (2001)
17. Igel, C., Glasmachers, T., Heidrich-Meisner, V.: Shark. J. Mach. Learn. Res. **9**:993–996 (2008)
18. Jain, A.K., Duin, R.P.W., Mao, J.: Statistical pattern recognition: a review. IEEE Trans. Pattern Anal. Mach. Intell. **22**(1), 4–37 (2000)
19. Japkowicz, N., Stephen, S.: The class imbalance problem: a systematic study. Intell. Data Anal. **6**(5), 429–449 (2002)
20. Julier, S.J., Uhlmann, J.K.: Unscented filtering and nonlinear estimation. In: Proceedings of the IEEE, pp. 401–422 (2004)
21. Leo, M., D'Orazio, T., Trivedi, M.: A multi camera system for soccer player performance evaluation. In: Proceedings of the ACM International Conference on Distributed Smart Cameras, pp. 1–8. (2009)
22. Lin, Y., Lee, Y., Wahba, G.: Support vector machines for classification in nonstandard situations. Mach. Learn. **46**(1), 191–202 (2002)
23. Liu, J., Tong, X., Li, W., Wang, T., Zhang, Y., Wang, H.: Automatic player detection, labeling and tracking in broadcast soccer video. Pattern Recognit. Lett. **30**(2), 103–113 (2009)
24. Poppe, C., Bruyne, S.D., Verstockt, S., de Walle, R.V.: Multi-camera analysis of soccer sequences. In: Proceedings of the IEEE Conference on Advanced Video and Signal Based Surveillance, pp. 26–31, (2010)
25. Prosser, B., Gong, S., Xiang, T.: Multi-camera matching using bi-directional cumulative brightness transfer functions. In: Proceedings of the British Machine Vision Conference, pp. 64.1–64.10, (2008)
26. Ramanath, R., Snyder, W.E., Bilbro, G.L., Sander, W.A.: Demosaicking methods for Bayer color arrays. J. Electron. Imaging **11**, 306–315 (2002)
27. Ren, J., Xu, M., Orwell, J., Jones, G.A.: Multi-camera video surveillance for real-time analysis and reconstruction of soccer games. Mach. Vision Appl. **21**, 855–863 (2010)
28. Rifkin, R., Klautau, A. In defense of one-vs-all classification. J. Mach. Learn. Res. **5**,101–141 (2004)
29. Muñoz, Salinas, R.: A bayesian plan-view map based approach for multiple-person detection and tracking. Pattern Recognit. **41**(12), 3665–3676 (2008)
30. Schlipsing, M., Salmen, J., Igel, C.: Echtzeit-Videoanalyse im Fußball-Entwurf eines Live-Systems zum Spieler-Tracking. Künstliche Intelligenz **27**(3), 235–240 (2013)
31. Teichman, A., Thrun, S.: Group induction. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), (2013)
32. Turk, M., Pentland, A.: Eigenfaces for recognition. J. Cogn. Neurosci. **3**(1), 71–86 (1991)
33. Vandenbroucke, N., Macaire, L., Postaire, J.G.: Color image segmentation by pixel classification in an adapted hybrid color space: application to soccer image analysis. Comput. Vis. Image Underst. **90**(2), 190–216 (2003)
34. Vapnik, V.: Statistical Learning Theory. Wiley (1998)
35. Xinguo, Y., Farin, D.: Current and emerging topics in sports video processing. In: Proceedings of the IEEE International Conference on Multimedia and Expo, pp. 526–529 (2005)
36. Xu, M., Orwell, J., Jones, G. Tracking football players with multiple cameras. In: Proceedings of the International Conference on Image Processing, vol. 5, pp. 2909–2912, (2004)
37. Zhang, N., Duan, L.Y., Li, L., Huang, Q., Du,J., Gao, W., Guan, L.: A generic approach for systematic analysis of sports videos. ACM Trans. Intell. Syst. Technol. **3**(3), 46:1–46:29 (2012)
38. Zhang, Z.: A flexible new technique for camera calibration. IEEE Trans. Pattern Anal. Mach. Intell. **22**(11), 1330–1334 (2000)
39. Zivkovic, Z., Kröse, B.: An EM-like algorithm for color-histogram-based object tracking. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 798–803, (2004)

Marc Schlipsing received his Master degree in Advanced Methods of Computer Science from Queen Mary College (University of London) in 2006 and his Diploma in Applied Computer Science from the Technical University Dortmund, Germany, in 2007. Starting his Ph.D. studies at the Institut für Neuroinformatik, Ruhr-Universität Bochum (RUB), in 2008, he is now head of the Real-time Computer Vision research group. His research interests include computer vision algorithms for advanced driver assistance systems and video-based sports analysis.

Jan Salmen earned his Diploma in computer science from the Technical University Dortmund in 2006. Since then, he worked as research associate at the Institut für Neuroinformatik. His main research interests are video-based object detection, feature learning and automatic optimization of computer vision algorithms. From 2009 until 2013, he was the head of the research group Real-time Computer Vision and received his Ph.D. in 2013.

Marc Tschentscher studied Applied Computer Science at the Ruhr-Universität Bochum, received his Bachelor degree in 2010 and his Master in 2012. Since then he is a research associate in the aforementioned Real-time Computer Vision Group.

Christian Igel studied Computer Science at the Technical University Dortmund. In 2002 he finished his Ph.D. studies at the Faculty of Technology, Universität Bielefeld, and qualified as a professor at the Faculty of Electrical Engineering and Information Technology (RUB) in 2010. From 2002 until 2010 he was junior professor for Optimization of Adaptive Systems at the Institut für Neuroinformatik (RUB). Since 2010 he is a Professor at the Department of Computer Science (DIKU) at the University of Copenhagen.